

Session 2

Preliminary to Software Testing

xfzhang@suda.edu.cn

Software Testing and Quality Assurance

Objectives

- ◆ Review of Session 1
 - ◆ Why we need SQA
 - ◆ Definition of SQA
 - ◆ QA VS. QC
 - ◆ Testing overview: what, who, when

Software Testing and Quality Assurance

◆ A question:

Testing is a subset of quality assurance or
quality control?

Software Testing and Quality Assurance

Objectives

- ◆ In this session, you will learn:
 - ◆ Comparison of some concepts
 - ◆ Testing life cycle
 - ◆ Model of testing
 - ◆ Types of testing

Software Testing and Quality Assurance

Comparison of some concepts

- ◆ What is “defect”?
- ◆ Verification / Validation
- ◆ Testing / Debugging
- ◆ White-box testing / Black-box testing

Software Testing and Quality Assurance

First Bug- 1947

Harvard University Mark II Aiken Relay Calculator

9/9

0800 Antcom started

1000 stopped - antcom ✓

1300 032 MP - MC

032 PRO 2

conv

Relays 6-2 in 033 failed special speed test in relay

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi-Adder Test.

1545

Relay #70 Panel F (moth) in relay.

First actual case of bug being found.

1630 Antcom started.

1700 closed down.



Photo # NH 96919-KN Commodore Grace M. Hopper, 1984



http://en.wikipedia.org/wiki/Grace_Hopper

Software Testing and Quality Assurance

What is “defect”?

- **Fault** (故障) : occurs in the process of **writing** a program.
- **Error** (错误) : is the manifestation of one or more errors. (**internal** state)
- **Failure** (失效) : occurs when faulty piece of code is executed leading to an incorrect state that propagates to the program's **output**.
- **Incident** (事故) : **no message** is displayed when failure occurs.



Defect (Bug)

- PIE model
 - E: Execution 执行
故障代码被执行到
 - I: Infection 感染
– 产生错误的中间状态
 - P: propagation 传播
– 被感染的状态传播到系统外部被观察和发现

Software Testing and Quality Assurance

- void test(int iLength, char iData[])

```
0 {  
1   int x=0;  
2   int y=0;  
3   for(int loop=0; loop<iLength; loop++)  
4   {  
5       if("A"==iData[loop])  
6           x++;  
7       else  
8           if("a"==iData[loop])  
9               x++;  
10          else  
11              y++;  
12  }  
13  printf ("%d,%d\n", x, y);  
14 }
```

char iData[]="John"

char iData[]="Amy"

char iData[]="Anna"

y++;

fault – error -- failure

Software Testing and Quality Assurance

```
public static int numZero (int [ ] arr)
{ // Effects: If arr is null throw NullPointerException
  // else return the number of occurrences of 0 in arr
  int count = 0;
  for (int i = 1; i < arr.length; i++)
  {
    if (arr [ i ] == 0)
    {
      count++;
    }
  }
  return count;
}
```

Fault: Should start searching at 0, not 1

Test 1
[2, 7, 0]
Expected: 1
Actual: 1

Error: i is 1, not 0, on the first iteration
Failure: none

Test 2
[0, 2, 7]
Expected: 1
Actual: 0

Error: i is 1, not 0
Error propagates to the variable count
Failure: count is 0 at the return statement

Software Testing and Quality Assurance

Bug Kinds (diff. kinds of errors)

- ***Syntactic*** errors:
 - Mal-formed program:

```
int square(int x) {  
    return x*x  
} *** syntax error at line 2  
    ';' expected
```

- ***Semantic*** errors:
 - ***Symbol*** errors
 - ***Type*** errors
 - *Other* semantic errors:
(e.g. uninitialized vars)

```
int square(int x) {  
    return n*n;  
} *** symbol error at line 2  
    undefined variable "n"
```

```
int square(float x) {  
    return x*x;  
} *** type error at line 2  
    function returns float, not int
```

- ***Logical*** errors:
 - Compiler: "no errors"

```
int square(int x) {  
    return x+x;  
}  
  
no errors found!!!
```

Software Testing and Quality Assurance

What is “defect”?

- Software defect definition
 - All kinds of software problems
 - Exist in code, **data** and **documents**
 - Out of accord with user expectation

Software Testing and Quality Assurance

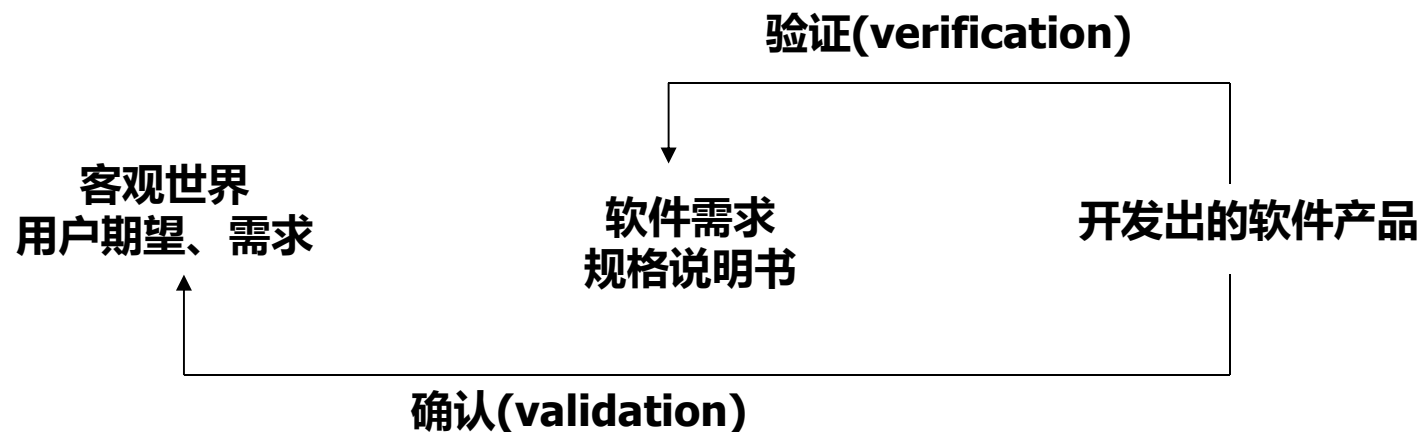
What is “defect”?

- Defect example:
 - Software function cannot be executed correctly
 - Shortcoming
 - running slowly
 - Inconsistency
 - Ctrl+S can't save all applications
 - User interface design defect
 - an button should show 5 words on it, but only 3 words could be seen

Software Testing and Quality Assurance

Verification vs. validation

- **Verification** (验证) : The software should conform to its specification
(Are we building the product right?)
- **Validation** (确认) : The software should do what the user really requires
(Are we building the right product?)



Software Testing and Quality Assurance

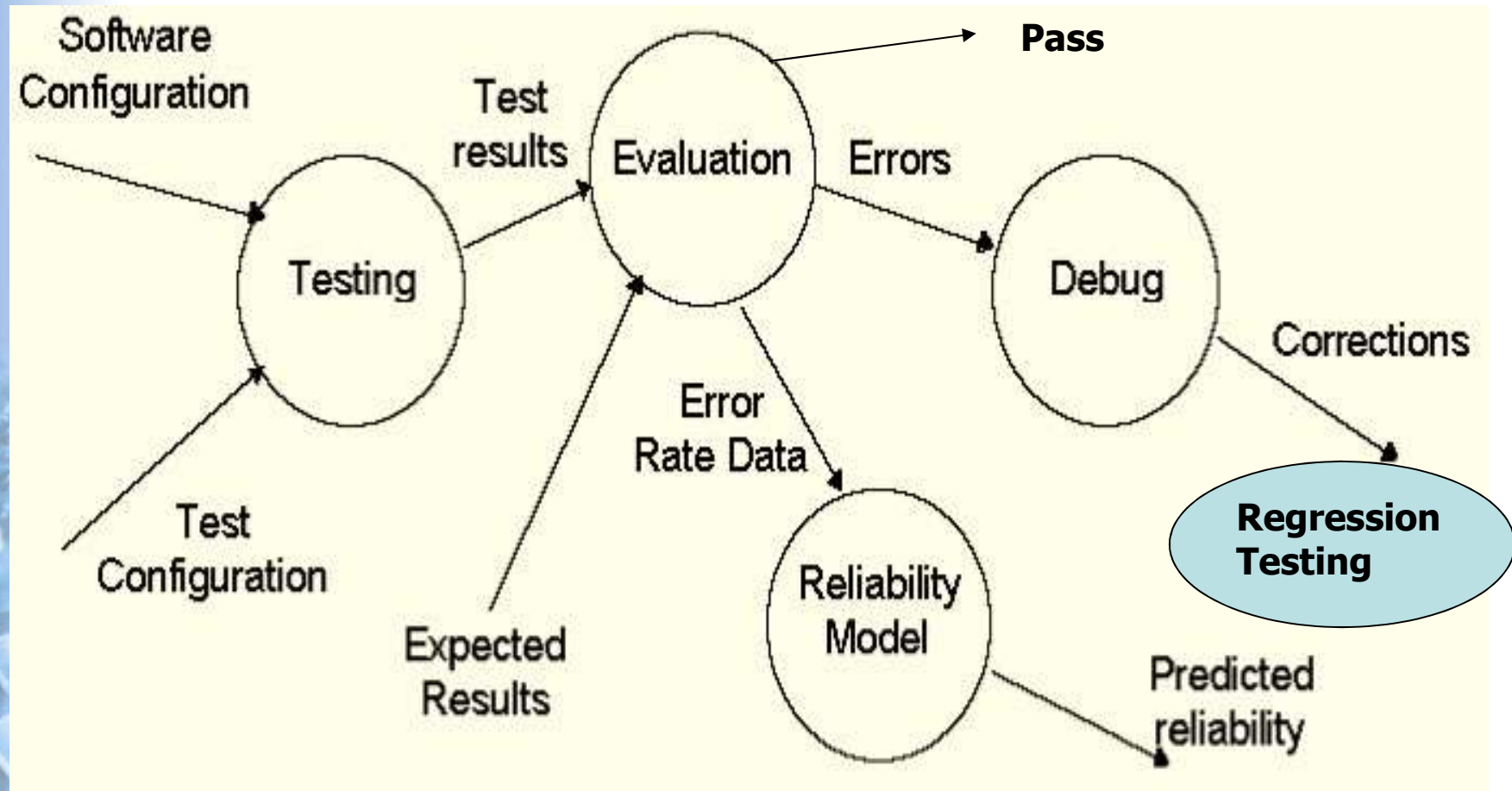
Verification vs. validation

**1 meaning;
2 purpose;
3 time;
4 participant;
5 activities**

S.N.	Verification	Validation
1	Are you building it right?	Are you building the right thing?
2	Ensure that the software system meets all the functionality.	Ensure that functionalities meet the intended behavior.
3	Verification takes place first and includes the checking for documentation, code etc.	Validation occurs after verification and mainly involves the checking of the overall product.
4	Done by developers.	Done by Testers.
5	Have static activities as it includes the reviews, walkthroughs, and inspections to verify that software is correct or not.	Have dynamic activities as it includes executing the software against the requirements.

Software Testing and Quality Assurance

Testing vs. Debugging



Test Information Flow

Software Testing and Quality Assurance

Testing vs. debugging

Testing :

It involves the identification of bug/defect in the software without correcting it.

Testing is performed in the testing phase.

Debugging :

It involves **identifying , isolating and fixing** the problems/bugs .

Debugging can be performed in the development phase while conducting Unit Testing or in phases while fixing the reported bugs.

Software Testing and Quality Assurance


White-box vs. Black-box Test

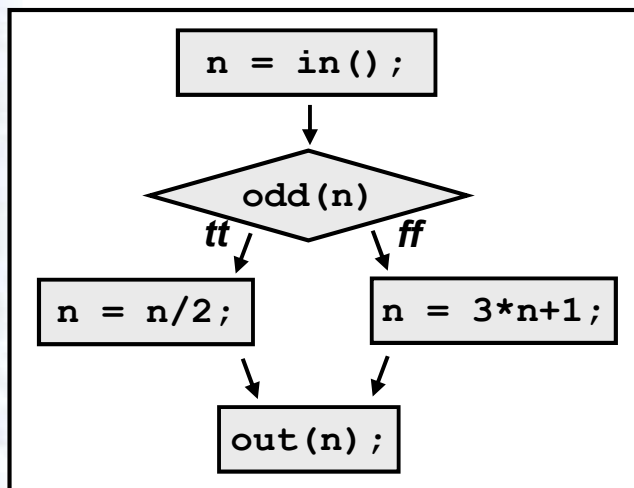
Test Case:


- To test whether a component conforms to project specification – **Black Box Testing**
- To test whether a component conforms to development design – **White Box Testing**

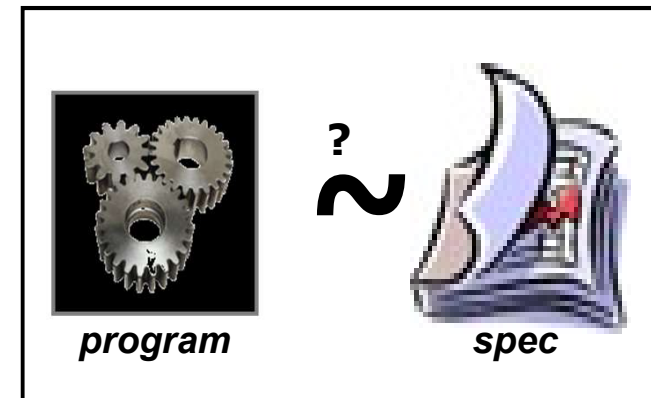
Software Testing and Quality Assurance

White-box vs. Black-box Test

-  **White-box Testing:**
 - *structural testing*
 - *internal testing*
- Test focus:
 - *source code / design*



-  **Black-box Testing:**
 - *behavioral testing*
 - *external testing*
 - *input-output testing*
- Test focus:
 - *specification* (or intention)



Complementary Approaches!!!

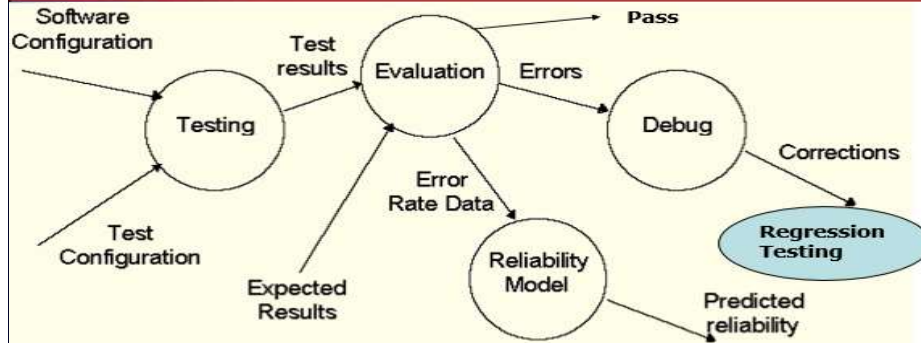
Software Testing and Quality Assurance

Objectives

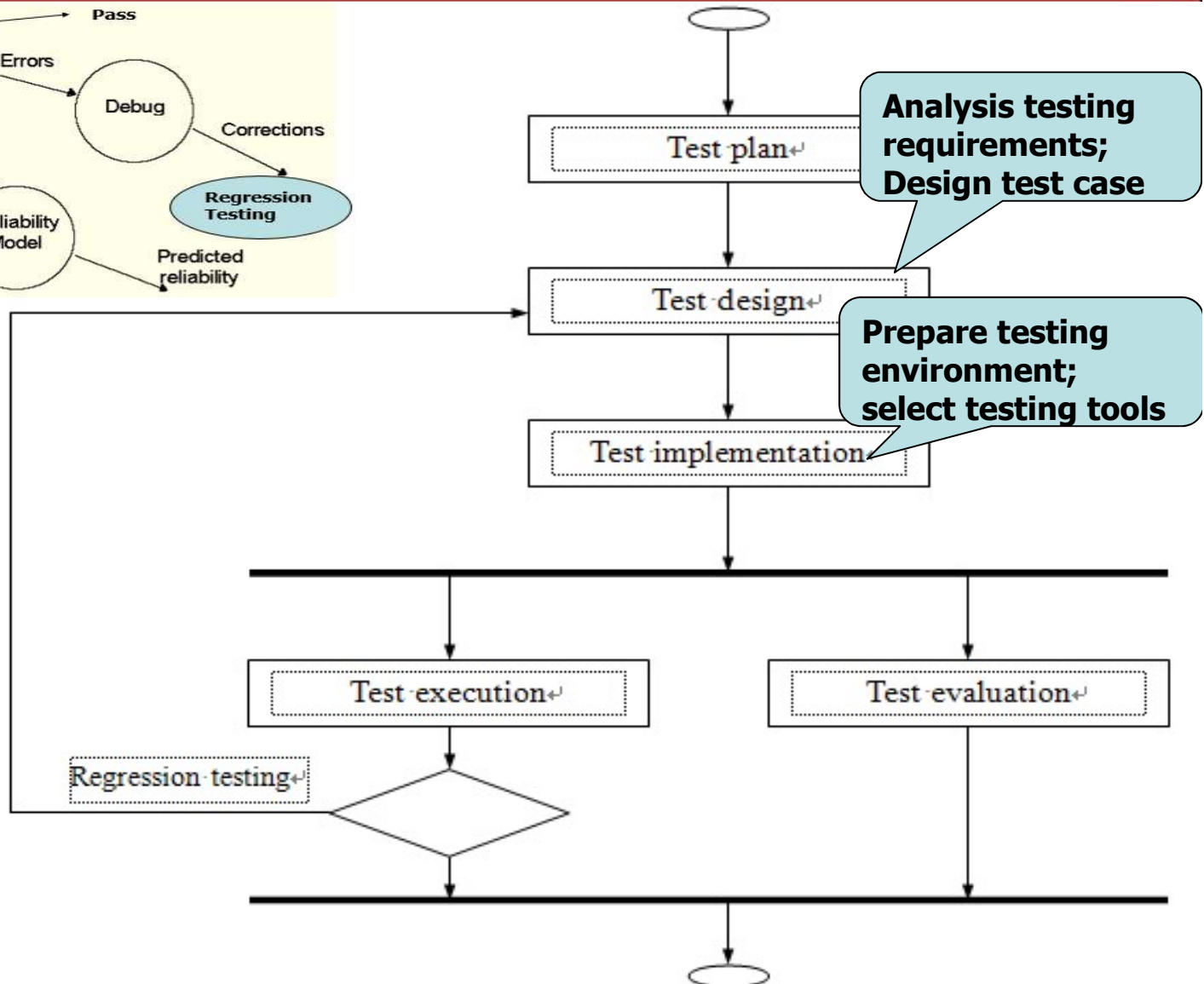
- ◆ In this session, you will learn:
 - ◆ Comparison of some concepts
 - ◆ Testing life cycle
 - ◆ Model of testing
 - ◆ Types of testing

Software Testing and Quality Assurance

Testing Life Cycle



PDCA



Software Testing and Quality Assurance

Objectives

- ◆ In this session, you will learn:
 - ◆ Comparison of some concepts
 - ◆ Testing life cycle
 - ◆ Model of testing
 - ◆ Types of testing

Software Testing and Quality Assurance

Model of testing

- ◆ The V model
- ◆ The W model

Software Testing and Quality Assurance

The V Model

- ◆ To reduce the cost of correcting defects, the defects must be detected early in the development life cycle.
- ◆ The earlier, the better.

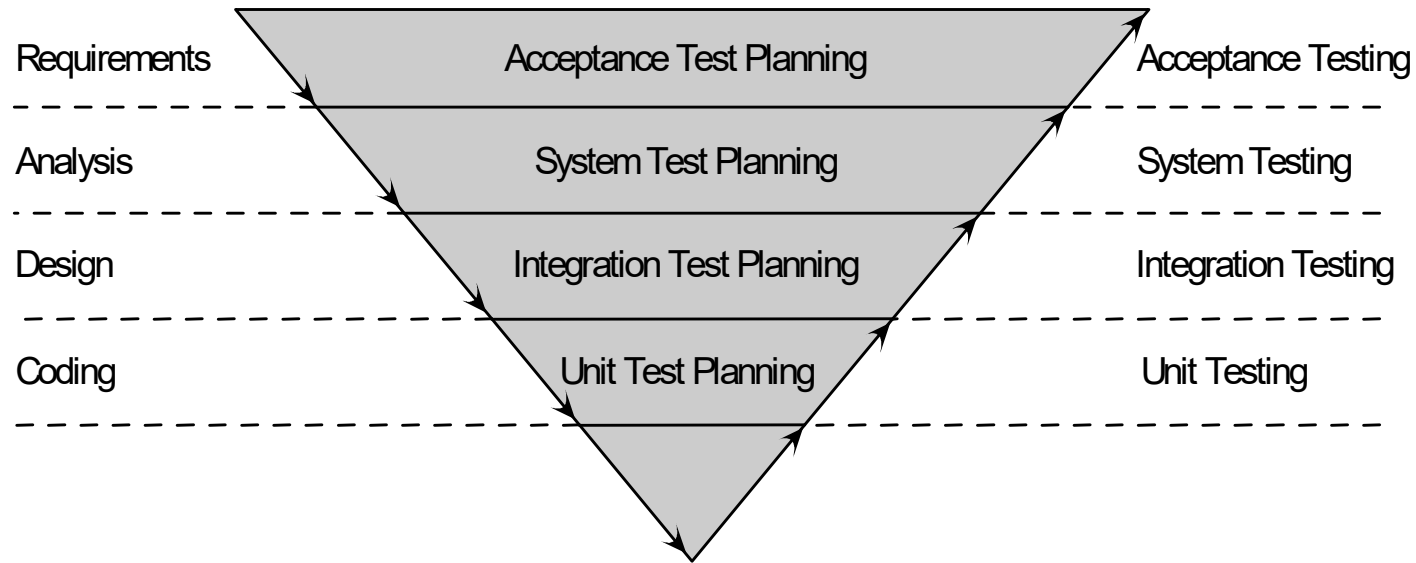
Software Testing and Quality Assurance

The V Model

- ◆ The V model proposes an approach to software development in which both the software development process and the software test process begin simultaneously.
- ◆ When the project starts, the development team starts the software development process and the **testing team starts planning** for the test process.
- ◆ This planning is **based on the documents** created during the development process.

Software Testing and Quality Assurance

The V Model (Contd.)



- ◆ The V model places the development phases such as requirements, analysis, design, and coding on one side of the V.
- ◆ The various types of testing such as unit, integration, system, and acceptance, are placed on the other side of the V.

Software Testing and Quality Assurance

The V Model (Contd.)

- ◆ What is the disadvantages of the V model?
- ◆ The V model focuses only on **dynamic testing** and does not mention the benefits and effectiveness of static testing techniques such as reviews, inspections, and static code analysis.

Software Testing and Quality Assurance

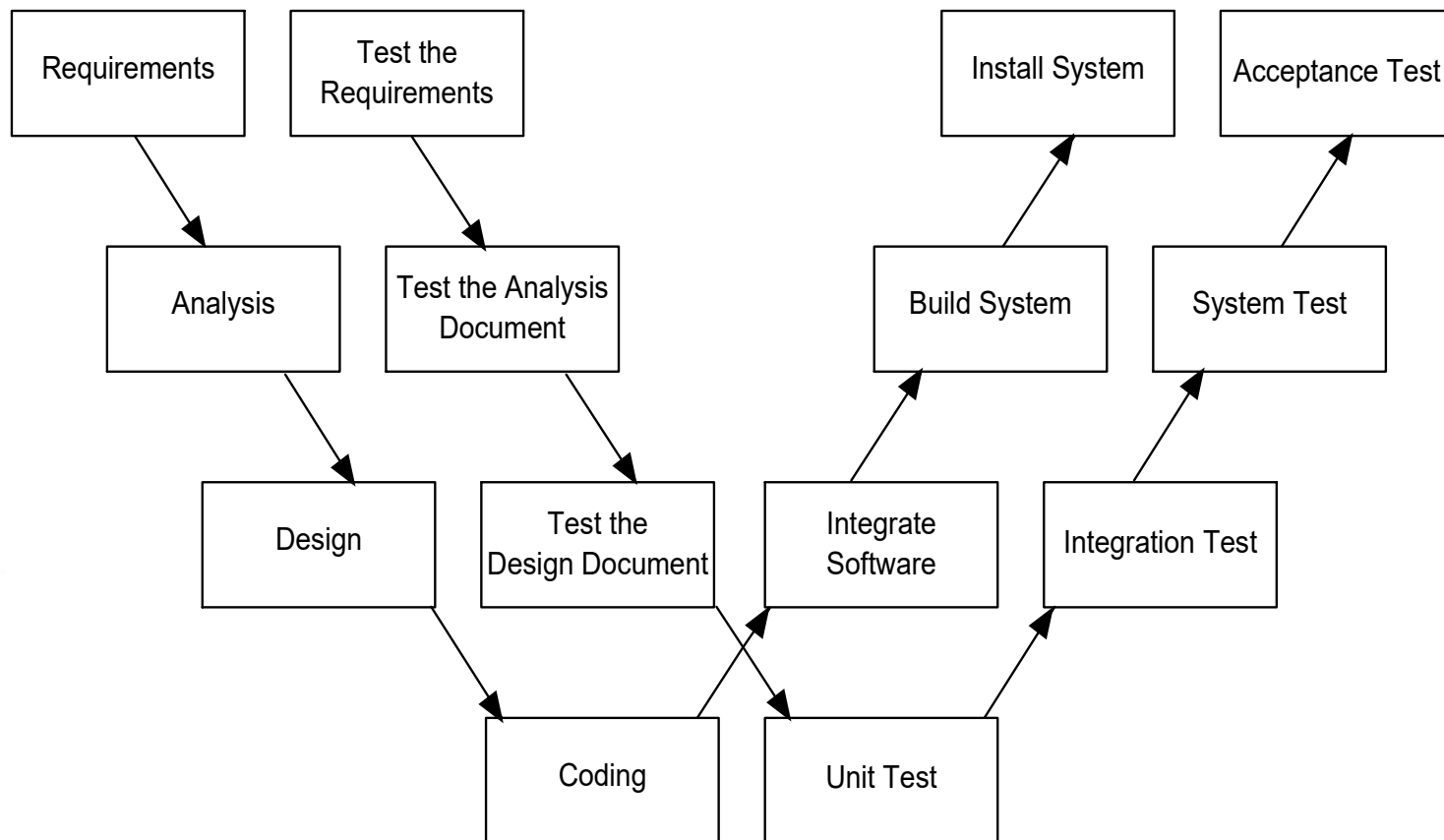
The W Model

- ◆ The W model is an extension of and addresses the limitations of the V model.
- ◆ It focuses on a testing methodology where **static testing** techniques are applied during the various stages of development. This results in tests being performed in parallel with the development process.
- ◆ These techniques **are cheaper and more effective** than dynamic testing.

Software Testing and Quality Assurance

The W Model (Contd.)

- ◆ The following figure shows the W model of testing.



Software Testing and Quality Assurance

The W Model

- ◆ The W model places the test process on an equal footing with the development process.
- ◆ The W model adopts both of static testing and dynamic testing.
- ◆ Recall verification & validation.

Software Testing and Quality Assurance

The V Model VS. W Model

Phase of the development life cycle	V model Test Activity	W model Test Activity
Requirements	Acceptance Test Plan	Test the requirements Acceptance Test Plan
High-Level Design	System Test Plan Integration Test Plan	Test the design documents System Test Plan Integration Test Plan
Detailed Design	Unit Test Plan	Test the design documents Unit Test Plan
Coding	Unit Testing	Unit Testing
Integration	Integration Testing System Testing	Integration Testing System Testing
Acceptance	Acceptance Testing	Acceptance Testing

Software Testing and Quality Assurance

Objectives

- ◆ In this session, you will learn:
 - ◆ Comparison of some concepts
 - ◆ Testing life cycle
 - ◆ Model of testing
 - ◆ Types of testing

Software Testing and Quality Assurance

Types of testing

- Types of testing
 - C1:Source of test generation
 - C2:Life cycle phase in which testing takes place
 - C3:Goal of a specific testing activity

Software Testing and Quality Assurance

Types of testing

- C1:Source of test generation

Artifact	Technique
Requirements (informal)	Black-box
Code	White-box
Requirements and code	Black-box and white-box
Formal model: graphical or mathematical specification	Model-based specification
Component's interface	Interface testing

Software Testing and Quality Assurance

Types of testing

- C2:Life cycle phase

Phase	Technique
Coding	Unit testing
Integration	Integration testing
System integration	System testing
Post system, pre-release	Alpha/Beta-testing /Acceptance testing
Maintenance	Regression testing

Software Testing and Quality Assurance

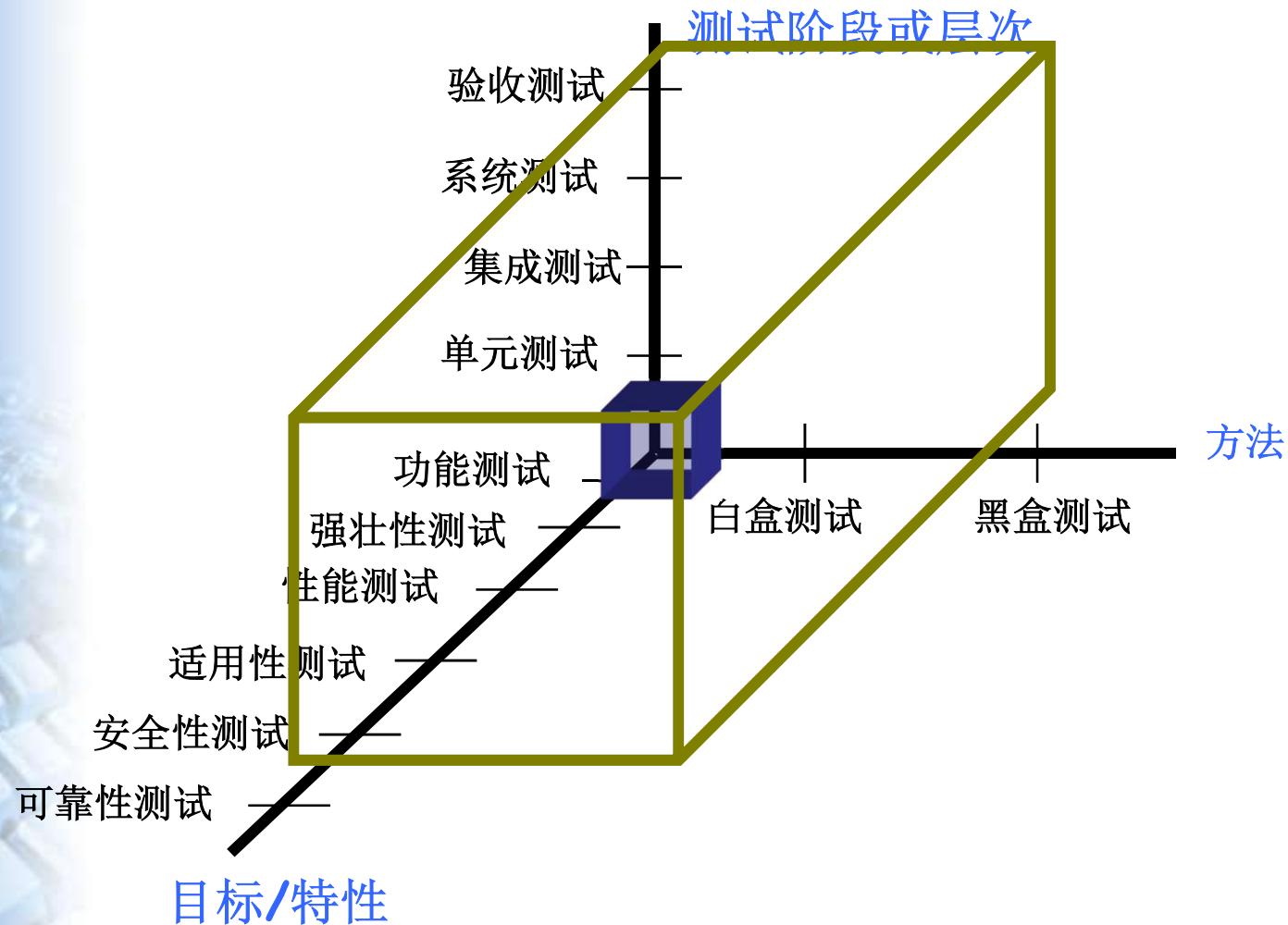
Types of testing

- C3:Goal-directed testing

Goal	Technique
Advertised features	Functional
Security	Security
Invalid inputs	Robustness
Errors in GUI	GUI
Operational correctness	Operational
Reliability assessment	Reliability
System performance	Performance
Customer acceptability	Acceptance
Business compatibility	Compatibility
Peripherals compatibility	Configuration
Foreign language compatibility	Foreign language

Software Testing and Quality Assurance

Types of testing



Software Testing and Quality Assurance

Summary

- ◆ In this session, you learned that:
 - ◆ Comparison of some concepts
 - ◆ Testing life cycle
 - ◆ Model of testing
 - ◆ Types of testing

Software Testing and Quality Assurance

◆ Questions:

1 Testing life cycle

2 V model and W model